

ตอนที่ 1: INTERNET AND WORLD WIDE WEB

รวม 10 คะแนน

1. **ARPANET (1 คะแนน)** เป็นโปรเจกต์แรกที่พยายามเชื่อมต่อหลายเครือข่ายเข้าด้วยกัน
2. ในปี 1983 โพรโทคอล **TCP/IP (1 คะแนน)** ถูกคิดค้นขึ้นมาแทนโพรโทคอล NCP เนื่องจากมีความสามารถและความคล่องตัวมากกว่า โพรโทคอลนี้ยังคงใช้อยู่ในปัจจุบัน
3. แต่ละเครือข่ายใน Internet ต้องสามารถรับส่งข้อมูลหากันได้ ดังนั้นแต่ละเครือข่ายต้องมีตัวแทนเพื่อเป็นศูนย์กลางในการรับส่งข้อมูล ซึ่งเราเรียกว่า **Host (1 คะแนน)**
4. แต่ละส่วนย่อยใน Internet ถูกควบคุมดูแลโดยกลุ่มที่แตกต่างกัน ไม่ว่าจะเป็นภาครัฐ องค์กรโทรศัพท์ หรือ มหาวิทยาลัย ดังนั้น **ไม่มีใคร (1 คะแนน)** เป็นเจ้าของ Internet
5. **Internet of Things (1 คะแนน)** เป็นการนำเอาเครื่องใช้ไฟฟ้าภายในบ้านมาเป็นอุปกรณ์ที่เชื่อมต่อกับ Internet ได้
6. Web Browser จะติดต่อกับ **Domain Name Server (1 คะแนน)** เพื่อทำการแปลง www.facebook.com ไปเป็น 69.63.176.13
7. จงบอกหน้าที่ของ Internet Protocol (2 คะแนน)
ควบคุมการไหลของข้อมูลจาก Host สู่ Host โดยใช้ IP Address เป็นตัวระบุต้นทางและปลายทาง
8. จงอธิบายลักษณะของข้อมูลในยุค Web 2.0 (2 คะแนน)
ข้อมูลส่วนมากมาจากผู้ใช้ เช่น โปสใน Facebook, Twitter, และ Instagram

ตอนที่ 2: MARKUP LANGUAGE

รวม 30 คะแนน

9. ในภาษา Hypertext Markup Language (HTML) **Hyperlink (1 คะแนน)** เป็นคำสั่งที่เชื่อมโยงแต่ละ HTML ไฟล์เมื่อมีการคลิกเมาส์

10. ในภาษา HTML5 มีการเพิ่ม **<video> (1 คะแนน)** ทำให้ไม่ต้องลง Adobe Flash หรือ Microsoft Silverlight

11. จงอธิบายลักษณะของภาษา Standard Generalized Markup Language (SGML) **(2 คะแนน)**
 เป็นภาษาที่แยกโครงสร้างหรือรูปแบบการแสดงผลออกจากข้อมูล โดยโครงสร้างหรือรูปแบบการแสดงผลจะถูกเขียนลงใน Document Type Definition

12. จงบอกความแตกต่างระหว่าง SGML และ HTML มาสองข้อ **(2 คะแนน)**
 - 1) SGML สามารถเขียน DTD เองได้ แต่ HTML มี DTD ที่ถูกกำหนดไว้แล้ว
 - 2) SGML สามารถใช้ได้กับเอกสารทุกรูปแบบ แต่ HTML ใช้ได้กับเอกสารบนเว็บเท่านั้น

13. จงบอกหน้าที่ของแต่ละภาษาใน HTML5 Family **(3 คะแนน)**
 - 1) HTML5 ใช้กำหนดโครงสร้างของหน้าเว็บ
 - 2) CSS ใช้กำหนดรูปลักษณ์ของหน้าเว็บ
 - 3) JavaScript ใช้กำหนดการตอบสนองจากผู้ใช้งาน

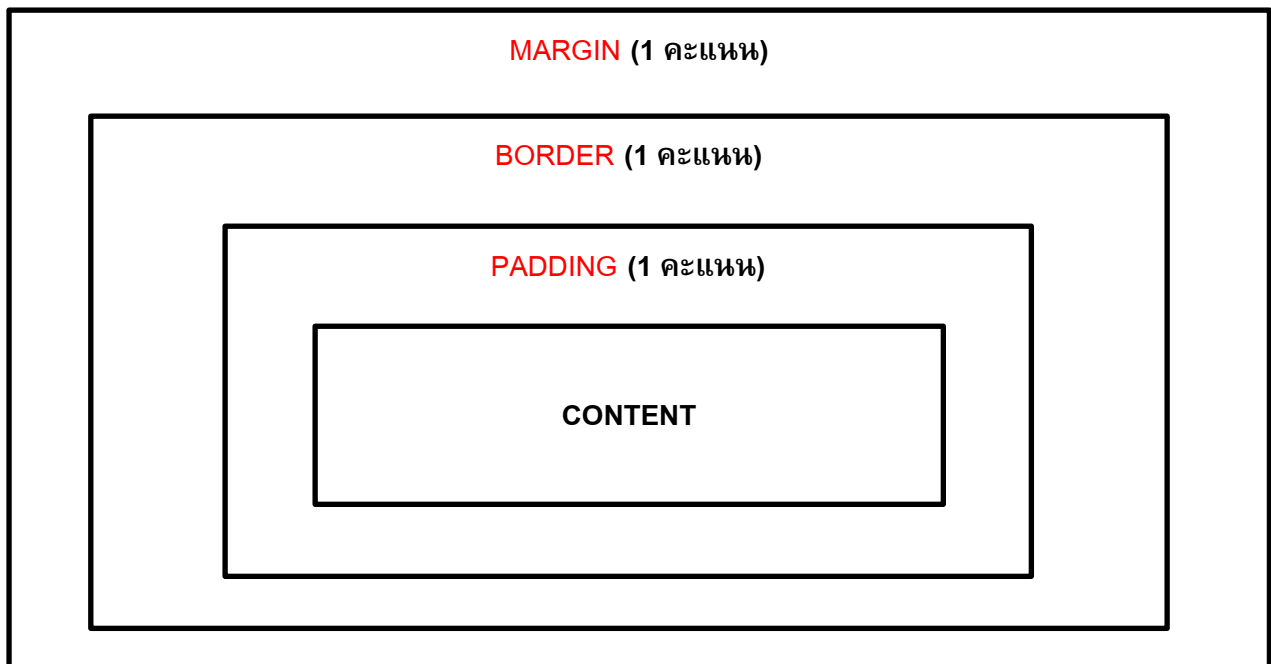
14. จาก XML ที่กำหนดให้ จงเขียน XML Tree **(3 คะแนน)**

<pre><?xml version="1.0" encoding="UTF-8"?> <main> <content> <one></one> <two></two> </content> <content> <four><four> </content> </main></pre>	<pre> graph TD Main([Main]) --> Content([Content]) Content --> One([One]) Content --> Two([Two]) Content --> Four([Four]) </pre>
---	--

15. จงบอกประโยชน์ของการใช้ Header Tag (1 คะแนน)
นอกจากใช้กำหนดหัวข้อแล้วยังถูก Search Engine นำไปใช้เป็น Keyword สำหรับการจัดลำดับ
16. จงบอกความแตกต่างระหว่างการใช้ href และ routerLink ในการสร้าง Link (2 คะแนน)
Link ที่ใช้ href จะทำการโหลดหน้าเว็บใหม่ทั้งหมด แต่ Link ที่ใช้ routerLink จะอัปเดตเฉพาะส่วนที่กำหนด
17. เพราะเหตุใดเราควรกำหนดค่าของ Attribute alt ใน เสมอ (1 คะแนน)
ข้อความใน alt จะถูกแสดงหากรูปภาพไม่ปรากฏ นอกจากนี้แล้วยังถูกอ่านออกเสียงใน Screen Reader สำหรับผู้พิการทางสายตา
18. จงบอกความแตกต่างระหว่าง Block-level Element และ Inline-level Element มาสองข้อ (2 คะแนน)
1) Block-level Element จะขึ้นบรรทัดใหม่เสมอ แต่ Inline-level Element จะไม่มีการขึ้นบรรทัดใหม่
2) Block-level Element กินพื้นที่เต็มความกว้างของจอ แต่ Inline-level Element ใช้แค่พื้นที่ที่จำเป็น
19. หากต้องการวาดรูปใน HTML โดยไม่ต้องการให้ภาพแตกเมื่อมีการซูม เราควรเขียนด้วย Tag SVG (1 คะแนน)
20. จงยกตัวอย่าง Tag ที่เกี่ยวข้องกับแบบฟอร์มและเพิ่งเข้ามาใหม่ใน HTML5 พร้อมอธิบายลักษณะการแสดงผลของ Tag นั้นมาสอง Tag (2 คะแนน)
1) type="color" ให้ผู้ใช้เลือกสีจากเมนู
2) type="date" ให้ผู้ใช้เลือกวันที่จากปฏิทิน
และอื่น ๆ
21. จงยกตัวอย่างการเรียกใช้งาน Cascading Style Sheet ในไฟล์ HTML มาสองรูปแบบ (2 คะแนน)
1) สร้างไฟล์ .css แล้วเรียกใช้งานด้วย <link>
2) เขียน CSS ภายใน <style>
และอื่น ๆ

22. จากเรียงลำดับส่วนต่าง ๆ ใน CSS Box Model

(3 คะแนน)



23. จงแปลง CSS ที่กำหนดให้เป็นรูปแบบ Atomic Stylesheet

(4 คะแนน)

<pre>.div1 { width: 300px; height: 100px; border: 1px solid blue; box-sizing: border-box; } .div2 { width: 300px; height: 100px; padding: 50px; border: 1px solid red; box-sizing: border-box; } .div3 { display: block; height: 100px; margin: 5px; border: 2px solid red; box-sizing: border-box; }</pre>	<pre>.w300 { width: 300px } .h100 { height: 100px } .b1 { border: 1px } .b2 { border: 2px } .bsolid { border: solid } .bblue { border: blue } .bred { border: red } .bsz { box-sizing: border-box } .p50 { padding: 50px } .db { display: block } .m5 { margin: 5px }</pre> <p>ชื่อกฎสามารถเปลี่ยนแปลงได้</p>
---	---

ตอนที่ 3: CLIENT-SERVER MODEL

รวม 10 คะแนน

24. จงยกตัวอย่าง Application ที่ควรใช้สถาปัตยกรรมแบบ Fat Client พร้อมให้เหตุผลประกอบ (2 คะแนน)
 Online Gaming เนื่องจากการ Render หน้าจอและข้อมูล Asset ทั้งหมดจะอยู่ในส่วน Client
 Mail Client เนื่องจากจัดรูปแบบข้อความและการแสดงรูปแบบนั้นจะอยู่ในส่วน Client
 (ตัวอย่างเดี๋ยวกี้เพียงพอ)
25. จงยกตัวอย่าง Application ที่ควรใช้สถาปัตยกรรมแบบ Fat Server พร้อมให้เหตุผลประกอบ(2คะแนน)
 Web Application เนื่องจาก Server ต้องสร้างเพจที่สำเร็จจาก Request
 Cloud Computing เนื่องจาก Server หลายเครื่องต้องประมวลผลร่วมกัน
 (ตัวอย่างเดี๋ยวกี้เพียงพอ)
26. จงบอกความแตกต่างระหว่าง Two-tier Model และ Three-tier Model (2 คะแนน)
 Two-tier Model ต่างจาก Three-tier Model ตรงที่ Application Logic ถูกแยกออกมาไว้ใน Application Server ดังนั้น Server จึงมีสองส่วนคือ Application Server และ Database Server
27. เพราะเหตุใดการส่ง HTTP Request ควรเป็นแบบ Asynchronous (1 คะแนน)
 เพราะ Web Browser สามารถ Render ส่วนอื่นของเว็บโดยไม่ต้องรอข้อมูลตอบกลับจาก Server
28. จงบอกลักษณะการทำงานของ Middleware ประเภท Publish/Subscribe (Push) (1 คะแนน)
 Server เฝ้าดูการทำงานของ Client และส่ง Response เมื่อ Client ว่าง
29. เพราะเหตุใด API-based Application จึงได้รับความนิยมมากในปัจจุบัน (1 คะแนน)
 API ใช้งานง่าย ล้มเหลวได้ยาก มีการกำหนดรายละเอียดข้อมูลที่รับส่งอย่างเป็นระบบ และสามารถรองรับการใช้งานหลายภาษา
30. จงบอกข้อดีของ Peer-to-Peer Model เมื่อเทียบกับ Client-Server Model (1 คะแนน)
 กระจายการทำงานให้กับทุกคอมพิวเตอร์ในเครือข่าย
 ขยายขนาดเครือข่ายได้ง่าย โดยไม่เสียประสิทธิภาพ
 และอื่น ๆ

ตอนที่ 4: WEB SERVER

รวม 10 คะแนน

31. ในปัจจุบันการเขียน Web Server เองเป็นสิ่งที่ไม่ค่อยมีใครทำแล้ว ส่วนมากจะใช้ **API (1 คะแนน)** เป็นสื่อกลางระหว่าง Web Client และ Web Server
32. Web Content คือข้อมูลที่อยู่ในรูปแบบของ **MIME (1 คะแนน)** โดยที่ Header ของ Web Content จะต้องระบุประเภทของข้อมูล
33. Content ที่ส่งกลับมาจาก Server แบบ **Static (1 คะแนน)** จะถูก Search Engine นำไปพิจารณาเมื่อมีการจัดลำดับผลลัพธ์ของเว็บ
34. จงอธิบายลักษณะการเชื่อมต่อแบบ Event-Driven Approach ใน Web Server (1 คะแนน)
สร้าง Event Handler มารอการตอบรับจากผู้ใช้ เมื่อมีการตอบรับจะทำคำสั่งใน Callback Function เมื่อทำเสร็จจะสร้าง Event Handler ใหม่มาทดแทน
35. จงบอกส่วนประกอบของ Request Line ใน HTTP Request (1 คะแนน)
<method> <uri> <version>
36. จงยกตัวอย่างการกระทำของ HTTP Request พร้อมอธิบายลักษณะของการกระทำนั้นมาสองตัวอย่าง (2 คะแนน)
- 1) GET เรียกดู Resource ใน Server
 - 2) POST ส่งข้อมูลไปแก้ไข Resource ใน Server
- และอื่น ๆ
37. การเชื่อมต่อใน HTTP 3.0 มีลักษณะอย่างไร และสามารถแก้ไขปัญหาที่เกิดขึ้นกับการเชื่อมต่อใน HTTP 2.0 ได้อย่างไร (3 คะแนน)
HTTP 3.0 ใช้การเชื่อมต่อแบบ QUIC ซึ่งเร็วกว่า TCP ของ HTTP 2.0 ข้อมูลที่ส่งแบบ QUIC จะถูกนำมาประมวลผลเลย ไม่รอข้อมูลที่เหลือ เพื่อป้องกันการเกิด Head-of-line Blocking

ตอนที่ 5: SINGLE PAGE APPLICATION

รวม 10 คะแนน

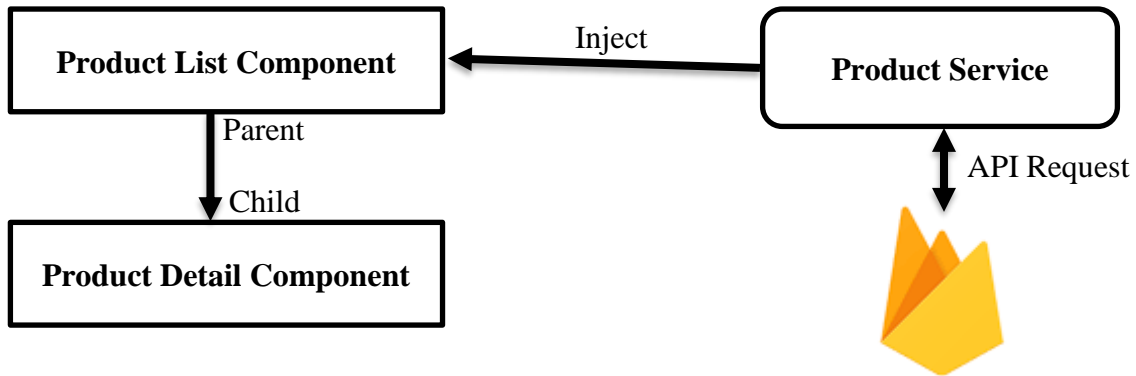
38. จงเขียนข้อดีและข้อเสียของวิธีการพัฒนา Web Application ทั้งสามแบบมาอย่างละสองข้อ (10คะแนน)

	ข้อดี	ข้อเสีย
Single Page Application	<ul style="list-style-type: none"> ➤ ให้ความรู้สึกเหมือนใช้งาน Mobile Application ➤ ไม่เสียเวลา Reload เพจ 	<ul style="list-style-type: none"> ➤ เพจสำเร็จจะสร้างเสร็จใน Browser ซึ่ง Search Engine Optimization จะไม่รอการประมวลผล ➤ ประสิทธิภาพขึ้นอยู่กับ Client ด้วย
Dynamic Website	<ul style="list-style-type: none"> ➤ เหมาะสำหรับ Search Engine Optimization เพราะ Server ส่ง เพจสำเร็จกลับมา ➤ Server ควบคุมการประมวลผลส่วนใหญ่ 	<ul style="list-style-type: none"> ➤ ทุกเพจใหม่จะต้อง Reload เต็มรูปแบบ ➤ แบ่งการพัฒนาเป็น Front-end Back-end ได้ยาก
Static Website	<ul style="list-style-type: none"> ➤ เหมาะสำหรับ Search Engine Optimization เพราะ Server ส่งเพจสำเร็จกลับมา ➤ ประสิทธิภาพคงที่เพราะไม่มีการสร้าง HTML 	<ul style="list-style-type: none"> ➤ ไม่เหมาะกับข้อมูลที่เปลี่ยนแปลงอยู่ตลอดเวลา ➤ สามารถใช้ Generator ช่วยจัดการข้อมูลที่เปลี่ยนแปลงบ่อยได้ แต่การติดตั้งเป็นไปได้ยาก

ตอนที่ 6: ANGULAR

รวม 30 คะแนน

39. กำหนดโครงสร้างของ Web Application ดัง Diagram ด้านล่าง จงเขียนคำสั่งที่ขาดหายไป (30คะแนน)



กำหนด Product Class ดังนี้

```

export class Product {
  id : number; // ไอดีของสินค้า
  name : string; // ชื่อสินค้า
  desc : string; // คำอธิบายสินค้า
  price : number; // ราคาสินค้า
}
  
```

จงเขียน Product Service ให้สำเร็จ

```

import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Observable } from 'rxjs';

import { Product } from './product';

@Injectable()
export class ProductService {

  constructor( // Inject HttpClient
    private http : HttpClient (2 คะแนน)
  ) { }

  private baseUrl = 'api/products';

  getProducts(): Observable<Product[]> { // Send HTTP GET request
    this.http.get<Product>(this.baseUrl); (3 คะแนน)
  }
}
  
```

จงเขียน Product Detail Component ให้สำเร็จ

```
// product-detail.component.js

import { Component, OnInit, Input } from '@angular/core';
import { Product } from '../product';

@Component({
  selector: 'app-product-detail',
  templateUrl: './product-detail.component.html',
  styleUrls: ['./product-detail.component.css']
})
export class ProductDetailComponent implements OnInit {

  @Input() prod : Product;

  constructor() { }

  ngOnInit() { }

}
```

```
// product-detail.component.html

<div>
  <h2> <!-- ชื่อสินค้า -->
    {{ prod.name }} (2 คะแนน)
  </h2>
  <p> <!-- คำอธิบายสินค้า -->
    {{ prod.desc }} (2 คะแนน)
  </p>
  Price : <!-- ราคาสินค้า -->
    {{ prod.price | currency:"USD" }}(2 คะแนน)
</div>
```

ตัวอย่างผลลัพธ์

Wireless Controller

The feel, shape, and sensitivity of the dual analog sticks and trigger buttons have been improved to provide a greater sense of control, no matter what you play

Price : \$46.96

จงเขียน Product List Component ให้สำเร็จ

```
// product-list.component.js

import { Component, OnInit } from '@angular/core';
import { Product } from '../product';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent implements OnInit {

  constructor(
    private pServ : ProductService
  ) { }

  products : Product[];

  ngOnInit() {
    // ดึงค่าสินค้าทั้งหมดโดยเรียกใช้ฟังก์ชันใน Product Service
    this.pServ.getProducts().subscribe(
      val => this.products = val
    ) (3 คะแนน)
  }
}
```

```
// product-list.component.html

<!-- ลูปสินค้าแต่ละชิ้น -->
<div *ngFor="let product of products" (3 คะแนน)>
  <!-- ส่งสินค้าไปให้ Product Detail Component เพื่อแสดงผลต่อไป -->
  <app-product-detail [prod]="product"></app-product-detail> (3 คะแนน)
</div>
```